**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application.

1. (Currently Amended) A method, comprising:

performing the following by a dispatcher that dispatches method entry and/or exit points to a plug-in:

receiving from a classfile registration information comprising a class name and [[a]] different method names for more than one of said class's methods, wherein said class name is in the form of a character string and where each of said method names are in the form of a character string and wherein each of said methods ~~being~~ have been previously modified with at least one additional byte code instruction to cause, for its respective method, a plug-in module's handler method to provide output function treatment for said respective method; ~~and,~~

referring to a plug-in pattern to determine which of a plurality of plug-in modules are appropriate for each of said methods, said plug-in pattern listing for each of said plug-in modules those of said methods that are to be handled with its corresponding output function treatment[[.]]; and,

updating a dictionary to include information from said plug-in pattern; and,

basing a new numeric name for said class upon the order in which said

classfile has registered with respect to the registration of other classfiles and passing said new numeric name to said classfile.

2. (Original) The method of claim 1 wherein said class name is in the form of a character string.

3. (Original) The method of claim 2 further comprising sending to said classfile a numeric name for said class.

4. (Original) The method of claim 3 wherein said numeric name is based upon the order in which said classfile has registered with respect to the registration of other classfiles.

5. (Original) The method of claim 4 wherein said numeric name is an integer.

6. (Original) The method of claim 1 wherein each of said method names are in a character string format.

7. (Original) The method of claim 6 further comprising assigning a different numeric name for each of said methods.

8. (Original) The method of claim 7 further comprising basing said numeric names upon the order in which said character string method names are received during said receiving.

9. (Original) The method of claim 8 wherein each next method in said order is configured to have a numeric name equal to a fixed increment above the numeric name for its immediately preceding method in said order.

10. (Original) The method of claim 9 wherein each of said numeric names is an integer.

11. (Currently Amended) The method of claim 1 wherein ~~said class name is in the form of a character string and where each of said method names are in the form of a character string, said method further comprising basing a new numeric name for said class upon the order in which said classfile has registered with respect to the registration of other classfiles,~~ said method further comprising basing a new numeric name for each of said methods upon the order in which said character string method names where received during said receiving.

12. (Original) The method of claim 11 wherein each of said numeric names is an integer.

13. (Original) The method of claim 11 further comprising, for each of said methods, updating a dictionary to include an entry for said numeric class name and the applicable method's numeric name, said method further comprising configuring said entry to identify said applicable method's appropriate one or more plug-in modules.

14. (Original) The method of claim 1 wherein said receiving of registration information is in response to said classfile being loaded.

15. (Original) The method of claim 1 wherein at least one of said plurality of plug-in modules further comprise a handler method that performs a time recordation function.

16. (Original) The method of claim 1 wherein at least one of said plurality of plug-in modules further comprise a handler method that performs a parameter value recordation function.

17. (Original) The method of claim 1 wherein said plurality of plug-in modules further comprise a handler method that performs a output function that increments a counter on a per method basis.

18. (Original) The method of claim 1 wherein said registration information further comprises arguments for each of said methods.

19. (Currently Amended) A machine readable medium containing stored program code which when processed by a machine causes a method to be performed, said method comprising:

performing the following by a dispatcher that dispatches method entry and/or exit points to a plug-in:

receiving from a classfile registration information comprising a class name and [[a]] different method names for more than one of said class's methods, wherein said class name is in the form of a character string and where each of said method names are in the form of a character string and wherein each of said methods ~~being~~ have been previously modified with at least one additional byte code instruction to cause, for its respective method, a plug-in module's handler method to provide output function treatment for said respective method; ~~and,~~

referring to a plug-in pattern to determine which of a plurality of plug-in modules are appropriate for each of said methods, said plug-in pattern listing for each of said plug-in modules those of said methods that are to be handled with its corresponding output function treatment[[.]];

updating a dictionary to include information from said plug-in pattern; and,

basing a new numeric name for said class upon the order in which said classfile has registered with respect to the registration of other classfiles and passing said new numeric name to said classfile.

20. (Original) The machine readable medium of claim 19 wherein said class name

is in the form of a character string.

21. (Original) The machine readable medium of claim 20 further comprising sending to said classfile a numeric name for said class.

22. (Original) The machine readable medium of claim 21 wherein said numeric name is based upon the order in which said classfile has registered with respect to the registration of other classfiles.

23. (Original) The machine readable medium of claim 22 wherein said numeric name is an integer.

24. (Original) The machine readable medium of claim 19 wherein each of said method names are in a character string format.

25. (Original) The machine readable medium of claim 24 wherein said method further comprises assigning a different numeric name for each of said methods.

26. (Original) The machine readable medium of claim 25 wherein said method further comprises basing said numeric names upon the order in which said character string method names are received during said receiving.

27. (Original) The machine readable medium of claim 26 wherein each next method in said order is configured to have a numeric name equal to a fixed increment above the numeric name for its immediately preceding method in said order.

28. (Original) The machine readable medium of claim 27 wherein each of said numeric names is an integer.

29. (Currently Amended) The machine readable medium of claim 19 wherein ~~said class name is in the form of a character string and where each of said method names are in the form of a character string, said method further comprising basing a new numeric name for said class upon the order in which said classfile has registered with respect to the registration of other classfiles,~~ said method further comprising basing a new numeric name for each of said methods upon the order in which said character string method names where received during said receiving.

30. (Original) The machine readable medium of claim 29 wherein each of said numeric names is an integer.

31. (Original) The machine readable medium of claim 29 wherein said method further comprises, for each of said methods, updating a dictionary to include an entry for said numeric class name and the applicable method's numeric name, said method further comprising configuring said entry to identify said applicable method's appropriate one or more plug-in modules.

32. (Original) The machine readable medium of claim 19 wherein said receiving of registration information is in response to said classfile being loaded.

33. (Original) The machine readable medium of claim 19 wherein at least one of said plurality of plug-in modules further comprise a handler method that performs a time recordation function.

34. (Original) The machine readable medium of claim 19 wherein at least one of said plurality of plug-in modules further comprise a handler method that performs a parameter value recordation function.

35. (Original) The machine readable medium of claim 19 wherein said plurality of plug-in modules further comprise a handler method that performs a output function that increments a counter on a per method basis.

36. (Previously Presented) The machine readable medium of claim 19 wherein said registration information further comprises arguments for each of said methods.

37. (Currently Amended) A distributed statistical recording method, comprising:

a) performing the following by a dispatcher that dispatches method entry and/or exit points to a plug-in:

receiving from a classfile registration information comprising a class name and different method names for more than one of said class's methods, wherein said class name is in the form of a character string and where each of said method names are in the form of a character string and wherein each of said methods have been previously ~~being~~ modified with at least one additional byte code instruction to cause, for its respective method, a plug-in module's handler method to provide output function treatment for said respective method, ~~and,~~ referring to a plug-in pattern to determine which of a plurality of plug-in modules are appropriate for each of said methods, said plug-in pattern listing for each of said plug-in modules those of said method that are to be handled with its corresponding output function treatment, updating a dictionary to include information from said plug-in pattern, and, basing a new numeric name for said class upon the order in which said classfile has registered with respect to the registration of other classfiles and passing said new numeric name to said classfile; and,

b) executing a method from said classfile, said executing causing said method's output function treatment to register information concerning

said method; and,

c) translating said information to a format employed within a distributed statistical records ("DSR") system.

38. (Original) The distributed statistical recording method of claim 37 wherein said class name is in the form of a character string.

39. (Original) The distributed statistical recording method of claim 38 further comprising sending to said classfile a numeric name for said class.

40. (Original) The distributed statistical recording method of claim 39 wherein said numeric name is based upon the order in which said classfile has registered with respect to the registration of other classfiles.

41. (Original) The distributed statistical recording method of claim 40 wherein said numeric name is an integer.

42. (Original) The distributed statistical recording method of claim 37 wherein each of said method names are in a character string format.

43. (Original) The distributed statistical recording method of claim 42 further comprising assigning a different numeric name for each of said methods.

44. (Original) The distributed statistical recording method of claim 43 further comprising basing said numeric names upon the order in which said character string method names are received during said receiving.

45. (Original) The distributed statistical recording method of claim 44 wherein each next method in said order is configured to have a numeric name equal to a

fixed increment above the numeric name for its immediately preceding method in said order.

46. (Original) The distributed statistical recording method of claim 45 wherein each of said numeric names is an integer.